DETAILED ACTION

Response to Arguments

As to Applicant's argument that,

"Oliver makes clear that integer unit 110 and FPU 120 operate in accordance with clock speeds that are decoupled from each other. See, e.g., Oliver, col. 4, line 65 to col. 5, line 1 ("The present invention decouples the clock speed of integer unit 110 and FPU 120 using command and data queues (or reservation stations) in dispatch unit 123 and control logic in execution pipeline clock controller 205.") (emphasis added). In contrast, claim 1 specifies that "each of the downstream stage and the upstream stage of the graphics pipeline operates in accordance with the clock rate in the graphics processor core clock domain." As such, and in accordance with claim 1, an increase (or a decrease) in the clock rate would apply to both the downstream stage and the upstream stage of the graphics pipeline, rather than having clock speeds decoupled in the manner contemplated by Oliver. The deficiencies of Giemborek, Williams, and Oliver are not remedied by the disclosure of the remaining cited reference."

The Examiner contests that Oliver, at FIG. 2 and 36-47, describes, "FIG. 2 illustrates variable speed floating point unit 120 in greater detail according to one embodiment of the present invention. Circuit block 210 generally designates components of floating point unit 120 that operate at the full speed of the Input Clock signal. These components include instruction buffer 121, issue unit 122, dispatch unit 123, load/store unit 211, and register array 212. However, the clock speed of floating point unit (FPU) execution pipeline 124 is variable and is controlled by execution pipeline clock controller 205. The Output Clock signal from execution pipeline clock controller 205 is a variable percentage (up to 100%) of the Input Clock signal." Further, 4:65-5:22 "The present invention decouples the clock speed of the integer unit 110 and FPU 120 using command and data queues (or reservation stations) in dispatch unit 123 and control logic in execution pipeline clock controller 205. Execution pipeline clock controller 205 set the clock speed of FPU execution pipeline 124 as a function of the number and type of commands in the reservation stations in dispatch unit 123. This information is determined from Reservation Station Full Levels status signals received from dispatch unit 123 and an Integer Pipe Stall Instruction signal received from issue unit 122."

Thus, the Floating Point Unit 120 and the Integer Unit 110 are both considered to operate in accordance with the ‘Input Clock’ rate in the processor core clock domain, where the Integer Unit 110 and a portion of the Floating Point Unit 120 always operate at the ‘Input Clock’ rate, as described above, and a second portion, the Floating Point Unit Execution Pipeline 124, operates off of a clock rate that may be the same as the 'Input Clock' rate or may be a variable percentage of the 'Input Clock' rate (up to 100%), which is still considered operating in accordance with the ‘Input Clock’ rate in the processor core clock domain, as the clock rate that is a variable percentage of the ‘Input Clock’ rate is directly derived from the ‘Input Clock' rate.

The ‘processor core clock domain’ is not considered to be limited to one clock rate, and is further considered to potentially include multiple simultaneously executing clock rates, such as the 'Input Clock' and 'Output Clock' rates described in the cited portions above.

Further, increasing the clock rate of the floating point unit execution pipeline as needed is considered increasing the performance level in response to detecting an over-utilization condition in order to increase one of the clock rates in the processor core clock domain, and decreasing the clock rate of the floating point unit execution pipeline as needed is considered decreasing the performance level in response to detecting an under-utilization condition to decrease one of the clock rates in the processor core clock domain.

Claim Rejections - 35 USC § 112

The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

Claims 1, 21, and 25 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claims 1, 21, and 25 are amended to recite the limitation (or a similar variation of) "wherein each of the downstream stage and the upstream stage of the graphics pipeline operates in accordance with **the clock rate** in the graphics processor core clock domain". There is insufficient antecedent basis for this limitation in the claim.

The described graphics processor core clock domain may comprise multiple clock rates if, for example, the graphics processor core utilizes multiple clocks; thus, a graphics processor core clock domain isn't considered to inherently consist of a single clock rate, which is why 'the clock rate' has insufficient antecedent basis.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1 and 25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Giemborek et al. (US 6,950,105) in view of Williams et al. (US 6,397,343), and further in view of Oliver et al. (US 7,243,217).

As to claim 1, Giemborek describes a method of operating a graphics system having a sequence of at least two discrete performance levels with each performance level being defined by a core clock rate of a graphics processing unit and a memory clock rate, the method comprising:

operating the graphics system at the core clock rate and memory clock rate associated with a selected performance level, the corresponding clock rate being a minimum within supported clock rates to maintain the display rate within a normal range (column 1 lines 50-67 through column 2 lines 1-11 and Figure 1 describes graphics accelerator 10, which matches the speed of at least one of two or more clocks (e.g., engine clock 40 and memory clock 42) to levels (speeds) under software control to a rate sufficient to satisfy current display requirements. The graphics accelerator includes 2D/3D engine 20, overlay engine 22, and frame buffer 16. Further, column 7 lines 50-67 through column 8 lines 1-27 describes increasing the clock speeds of the clocks within the graphics accelerator 10 if the system is currently over-utilized and decreasing the clock speeds of the clocks within the graphics accelerator 10 if the system is currently under-utilized. Finally, column 1 lines 50-65 describes that the clock speeds are set such that display rate remains in a normal range (i.e., graphics display performance is not sacrificed). Thus, Giemborek is considered to describe operating the graphics system at the core clock rate and memory clock rate associated with a selected performance level the corresponding clock rate being a minimum within supported clock rates to maintain the display rate within a normal range).

Giemborek doesn't describe monitoring utilization of a graphics pipeline, the graphics pipeline being in the graphics processor core clock domain such that the performance level affects the clock rate of the graphics pipeline, wherein the graphics pipeline has a set of stages in which graphics data is processed in a pipelined sequence through each subsequent stage in the graphics pipeline, and wherein each of a downstream stage and an upstream stage of the graphics pipeline operates in accordance with the clock rate in the graphics processor core clock domain.

However, Williams describes monitoring the utilization of a graphics pipeline, the graphics pipeline being in the graphics processor core clock domain such that the performance level of the graphics system affects the clock rate of the graphics pipeline, wherein the graphics pipeline has a set of stages in which graphics data is processed in a pipelined sequence through each subsequent stage in the graphics pipeline, and wherein each of a downstream stage and an upstream stage of the graphics pipeline operates in accordance with the clock rate in the graphics processor core clock domain.

Williams describes a method and system that includes a device for dynamic graphics subsystem clock adjustment within a computer system having a CPU and a dedicated graphics subsystem. A system interface is coupled to the graphics subsystem to allow a controller to determine the graphics processing load placed on the graphics subsystem (column 4 lines 11-31 and column 6 lines 33-50). Williams further describes that monitoring said at least one attribute (in this case the graphics processing load placed on the graphics subsystem) comprises: monitoring at least one attribute indicative of utilization of a graphics pipeline within a graphics processor core clock domain and determining whether the graphic pipeline is under-utilized or over-utilized (column 6 lines 51-67, column 7 lines 1-20, and column 8 lines 11-22 describes that the device 100 determines the graphics subsystem load by monitoring the processing activity of graphics subsystem 200 via the pipeline control 206 (e.g., by snooping graphics commands and data flowing through a graphics pipeline to determine the activity level of the graphics pipeline) and adjusts the pipeline clock frequency accordingly),

wherein the graphics pipeline has a set of stages in which graphics data is processed in a pipelined sequence through each subsequent stage in the graphics pipeline (FIG. 2A and 6:59-67 "The exemplary graphics subsystem 200 includes a geometry unit 205, a scan conversion unit 204, and a texture unit 203. Graphics instructions and associated data are received and dispatched across a graphics subsystem bus 201. The graphics instructions and data are processed by the geometry unit 205, scan conversion unit 204, and texture unit 203, with a pipeline control 206 coordinating their operation and timing." Further, 8:10-22 "In addition to detecting the graphics subsystem load by monitoring the processing activity of graphics subsystem 200 via pipeline control 206 (e.g., by snooping graphics commands and data flowing through a graphics pipeline), it should be noted that device 100 can be configured to use additional inputs to determine graphics subsystem load."), and wherein each of a downstream stage and an upstream stage of the graphics pipeline operates in accordance with the clock rate in the graphics processor core clock domain (FIG. 1 and 6:13-50 "The device 100 of the present invention is comprised of a clock pulse generator 101 for generating a clock frequency. The clock frequency generated by the generator is variable over a broad frequency range. The clock frequency is coupled to a clock frequency output line 104. Clock frequency output line 104 is coupled to a graphics subsystem (e.g., graphics subsystem 200 of FIG. 2) and is used by the graphics subsystem to synchronize and pace its internal operations … [starting at 6:33] An interface 103 is coupled to the controller 102. The interface 103 is coupled to an interface line 106. The interface 103 transmits one or more input signals to the controller 102. The input signal transmitted to controller 102 depends upon the nature of the input device 100 relies upon to adjust the graphics subsystem clock frequency. For example, in a case where interface 103 is coupled to the graphics subsystem to receive an indication of the load on the graphics subsystem (e.g., the embodiment shown in FIG. 2A), this load indication is coupled to controller 102, which in turn, controls the graphics subsystem clock frequency accordingly. In this case, the device 100 dynamically adjusts the clock frequency in response to the load on the graphics subsystem. The controller adjusts the clock frequency generated by the clock pulse generator such that the clock frequency increases when the load on the graphics subsystem increases and the clock frequency decreases when the load on the graphics subsystem decreases." Further, FIG. 2A and 7:5-20 "The system interface line 106 is coupled to the pipeline control 206 of the graphics subsystem 200. The clock frequency line 104 is coupled to the geometry unit 205, the scan conversion unit 204, and the texture unit 203. The clock frequency serves as the master clock signal upon which the operation and synchronization of the components of the graphics subsystem 200 are derived. The device of the present invention 100 determines a load on the graphics subsystem via the activity detected in pipeline control 206 through the system interface line 106. In response to the determined load, the device 100 sets the clock frequency. When the load detected increases, the device 100 increases the clock frequency. When the load decreases, the device 100 decreases the clock frequency."

Thus, each of the at least one downstream stage and at least one upstream stage of the graphics pipeline (e.g., stages 203-205) operates in accordance with the clock rate (i.e., the clock pulse generator 101 generates a clock frequency) in the graphics processor core clock domain).

All the above-described limitations of claim 1 are known in Giemborek and Williams, the only difference is the combination of old elements into a single system and method.

Thus, it would have been obvious to one of ordinary skill in the art at the time of the invention to include in Giemborek the system and method of determining the processing load placed on a graphics subsystem by monitoring the level of activity of a graphics pipeline, and adjusting the frequency of the pipeline clock according to the determined load, the graphics pipeline being in the graphics processor core clock domain such that the performance level of the graphics system affects the clock rate of the graphics pipeline, wherein the graphics pipeline has a set of stages in which graphics data is processed in a pipelined sequence through each subsequent stage in the graphics pipeline, and wherein each of a downstream stage and an upstream stage of the graphics pipeline operates in accordance with the clock rate in the graphics processor core clock domain, as taught by Williams, as this doesn't change the operation of the rest of the system, and it could be used to achieve the predictable results of improving the efficiency of the 2D/3D graphics engine disclosed in Giemborek by monitoring the pipeline activity within the graphics engine and determining a required clock rate to be passed to the engine based, in part, on the monitored activity. One advantage of passing the graphics engines a variable clock rate based at least in part on calculated activity within a pipeline within the engine is that the system can further reduce its power consumption and optimize its calculated clock rates by adding the activity of the pipeline to the list of factors that are used when determining the clock rates.

Giemborek in view of Williams doesn't describe monitoring a graphics pipeline in a graphics processor core clock domain and detecting a percentage of clock cycles for which a downstream stage of the graphics pipeline is held up waiting for data inputs from an upstream stage of the graphics pipeline as an indicator of utilization and determining whether the graphics pipeline is under-utilized or over-utilized, wherein each of the downstream stage and the upstream stage of the graphics pipeline operates in accordance with the clock rate in the graphics processor core clock domain; and increasing the performance level in response to detecting an over-utilization condition in order to increase the clock rate in the graphics processor core clock domain and decreasing the performance level in response to detecting and under-utilization condition to decrease the clock rate in the graphics processor core clock domain.

However, Oliver describes a system and method including detecting a percentage of clock cycles within a processor core clock domain for which a downstream stage of a pipeline is held up waiting for data inputs from an upstream stage of the pipeline as an indicator of utilization and determining whether the pipeline is under-utilized or over-utilized (4:65-5:15 “The present invention decouples the clock speed of integer unit 110 and FPU 120 using command and data queues (or reservation stations) in dispatch unit 123 and control logic in execution pipeline clock controller 205. Execution pipeline clock controller 205 sets the clock speed of FPU execution pipeline 124 as a function of the number and type of commands in the reservation stations in dispatch unit 123. This information is determined from Reservation Stations Full Levels status signals received from dispatch unit 123 and an Integer Pipe Stall Instruction signal received from issue unit 122. Execution pipeline clock controller 205 sets the speed of the Output Clock signal to a high rate (Fast mode) if the reservation stations are filling up, if integer unit 110 is stalled waiting for the result from FPU 120, or if the commands in the reservation stations require multiple cycles to execute." Further, 5:40-62 discloses "Execution pipeline clock controller 205 increases the Output Clock signal speed as the level rises in each reservation station or if an opcode indicates that integer unit 110 is waiting for a result from FPU 120 (process step 420). Execution pipeline clock controller 205 also decreases the Output Clock signal speed as the level drops in each reservation station and if no queued opcode indicates that integer unit 110 is waiting for a result from FPU 120 (process step 425)."

Thus, when the percentage of clock cycles within the processor core clock domain for which the integer unit stage of the pipeline is held up waiting for inputs from the upstream floating point unit stage is greater than zero and the floating point unit is being clocked at the slower speed this acts as an indicator in determining that the pipeline is over-utilized (at which point an opcode is sent to the floating point unit indicating that its clock speed should be increased), and when the percentage of clock cycles at one or more points within the processor core clock domain for which the integer unit stage of the pipeline is held up waiting for inputs from the upstream floating point unit stage is not greater than zero and the floating point unit is being clocked at the faster speed this acts as an indicator in determining that the pipeline is under-utilized (at which point the clock speed of the floating point unit is reduced),

wherein each of the downstream stage and the upstream stage of the pipeline operates in accordance with the clock rate in the processor core clock domain (FIG. 2 and 36-47 “FIG. 2 illustrates variable speed floating point unit 120 in greater detail according to one embodiment of the present invention. Circuit block 210 generally designates components of floating point unit 120 that operate at the full speed of the Input Clock signal. These components include instruction buffer 121, issue unit 122, dispatch unit 123, load/store unit 211, and register array 212. However, the clock speed of floating point unit (FPU) execution pipeline 124 is variable and is controlled by execution pipeline clock controller 205. The Output Clock signal from execution pipeline clock controller 205 is a variable percentage (up to 100%) of the Input Clock signal." Further, 4:65-5:22 "The present invention decouples the clock speed of the integer unit 110 and FPU 120 using command and data queues (or reservation stations) in dispatch unit 123 and control logic in execution pipeline clock controller 205. Execution pipeline clock controller 205 set the clock speed of FPU execution pipeline 124 as a function of the number and type of commands in the reservation stations in dispatch unit 123. This information is determined from Reservation Station Full Levels status signals received from dispatch unit 123 and an Integer Pipe Stall Instruction signal received from issue unit 122."

Thus, the Floating Point Unit 120 and the Integer Unit 110 are both considered to operate in accordance with the 'Input Clock' rate in the processor core clock domain, where the Integer Unit 110 and a portion of the Floating Point Unit 120 always operate at the 'Input Clock' rate, as described above, and a second portion, the Floating Point Unit Execution Pipeline 124, operates off of a clock rate that may be the same as the 'Input Clock' rate or may be a variable percentage of the 'Input Clock' rate (up to 100%), which is still considered operating in accordance with the 'Input Clock' rate, where the 'Input Clock' rate is considered to be one of possibly multiple clock rates included in the processor core clock domain).

Further, increasing the clock rate of the floating point unit execution pipeline as needed is considered increasing the performance level in response to detecting an over-utilization condition in order to increase a clock rate in the processor core clock domain (in this case increasing the 'Output Clock' rate), and decreasing the clock rate of the floating point unit execution pipeline as needed is considered decreasing the performance level in response to detecting an under-utilization condition to decrease the clock rate in the processor core clock domain (in this case decreasing the 'Output Clock' rate).

All the elements of claim 1 are known in Giemborek in view of Williams and further in view of Oliver, the only difference is the combination of known elements into a single system and method.

Thus, it would have been obvious to one of ordinary skill in the art at the time of the invention to include in Giemborek in view of Williams the system and method of monitoring a graphics pipeline in a graphics processor core clock domain and detecting a percentage of clock cycles for which a downstream stage of a graphics pipeline is held up waiting for data inputs from an upstream stage of the graphics pipeline as an indicator of utilization and determining whether the graphics pipeline is under-utilized or over-utilized, wherein each of the downstream stage and the upstream stage of the graphics pipeline operates in accordance with the clock rate in the graphics processor core clock domain; and increasing the performance level in response to detecting an over-utilization condition in order to increase the clock rate in the graphics processor core clock domain and decreasing the performance level in response to detecting an under-utilization condition to decrease the clock rate in the graphics processor core clock domain, as suggested by Oliver, in order to achieve the predictable result of improving the efficiency and power conservation of the 2D/3D graphics engine (disclosed in Giemborek) by monitoring the pipeline activity within the graphics engine and determining an appropriate clock rate. The advantage of an adjustable clock rate is it allows the pipeline to meet processing demands when the required processing load is high while also conserving power when the required processing load is low.

As to claim 25, Giemborek describes a graphics system, comprising:

a graphics processor having a sequence of at least two discrete performance levels where each performance level is defined by a graphics processor core clock rate of the graphics processing unit and a memory clock rate (column 1 lines 50-67 through column 2 lines 1-11 and Figure 1 describes graphics accelerator 10, which matches the speed of at least one of two or more clocks (e.g., engine clock 40 and memory clock 42) to levels (speeds) under software control to a rate sufficient to satisfy current display requirements. The graphics accelerator includes 2D/3D engine 20, overlay engine 22, and frame buffer 16); and

a graphics memory coupled to said graphics processor by a graphics bus and operable at said memory clock rate (Figure 1 and column 2 lines 27-36 describes frame buffer memory 16);

the graphics system operating at the core clock rate and memory clock rate associated with the performance level selected by the performance level controller, the selected performance level being a minimum performance level capable of maintaining the display rate within a normal range (column 1 lines 50-65 describes that the clock speeds are selected to ensure that the graphics display performance is not degraded).

Giemborek doesn’t describe but Williams describes a graphics processor having a graphics pipeline in a graphics processor core clock domain, the single graphics pipeline having a set of stages in which graphics data is processed in a pipelined sequence through each subsequent stage in the graphics pipeline, wherein the graphics processor monitors the utilization of a graphics pipeline, the graphics pipeline being in the graphics processor core clock domain such that the performance level affects the clock rate of the graphics pipeline, and wherein a downstream stage and an upstream stage of the graphics pipeline both operate in accordance with the clock rate in the graphics processor core clock domain (see the corresponding section in the rejection of claim 1).

See the rejection of claim 1 for rationale to combine Williams with Giemborek.

Giemborek in view of Williams doesn't describe but Oliver describes a performance level controller, said performance level controller configured to monitor, as a function of time a percentage of clock cycles for which a downstream stage of the pipeline is held up waiting for data inputs from an upstream stage of the pipeline as an indicator of utilization and determining whether the pipeline is under-utilized or over-utilized (4:65-5:22 and 5:53-62 describes execution pipeline clock controller 205. Also see the corresponding section in the rejection of claim 1),

wherein the downstream stage and the upstream stage of the pipeline both operate in accordance with the clock rate in the graphics processor core clock domain (see the corresponding section in the rejection of claim 1); and

said performance level controller configured to increase said performance level to increase the clock rate in the processor core clock domain and to avoid over-utilization of said pipeline (5:53-56).

said performance level controller configured to decrease said performance level from a high performance level to a lower performance level to decrease the clock rate in the processor core clock domain to avoid under-utilization of said pipeline (5:57-62).

See the rejection of claim 1 for rationale to combine Oliver with Giemborek and Williams.

Claim 21 and 31 are rejected under 35 U.S.C. 103(a) as being unpatentable over Giemborek et al. (US 6,950,105) in view of Williams et al. (US 6,397,343), in view of Oliver et al. (US 7,243,217), and further in view of Gulick (US 6,061,802).

Concerning claim 21, Giemborek describes a method of operating a graphics system having a sequence of at least two discrete performance levels where each performance level is defined by a core clock rate of a graphics processing unit and a memory clock rate, the performance levels including a high performance level for processing complex three-dimensional graphical images and at least one lower power, lower performance level for processing less complex graphical images, the method comprising:

operating the graphics system at the core clock rate and memory clock rate associated with the selected performance level, the selected performance level being a minimum performance level sufficient to maintain the display rate within the normal range (see the corresponding section in the rejection of claim 1).

Giemborek doesn't describe but Williams describes monitoring the utilization of a single graphics pipeline in a graphics processor core clock domain, the single graphics pipeline having a set of stages in which graphics data is processed in a pipelined sequence through each subsequent stage in the graphics pipeline, the graphics pipeline being in the graphics processor core clock domain such that the performance level affects the clock rate of the graphics pipeline, wherein each of a downstream stage and an upstream stage of the graphics pipeline operates in accordance with the clock rate in the graphics processor core clock domain (see the corresponding section in the rejection of claim 1).

See the rejection of claim 1 for rationale to combine Williams with Giemborek.

Giemborek in view of Williams doesn't describe but Oliver describes monitoring a pipeline and detecting as a function of time a percentage of clock cycles that a downstream stage of the graphics pipeline is held up waiting for data inputs from an upstream stage of the pipeline as an indicator of utilization and determining whether the pipeline is under-utilized or over-utilized, wherein the downstream stage and the upstream stage of the pipeline both operate in accordance with the clock rate in the processor core clock domain (see the corresponding section in the rejection of claim 1);

in response to detecting a level of utilization greater than an over-utilization threshold for which a display rate of the graphics system is likely to be significantly decreased below a normal display rate, selecting a higher performance level to increase a clock rate in the graphics processor core clock domain (4:65-5:62, the variable speed clock applied to the floating point unit can easily be applied to one or more stages in a graphics processing pipeline (see 4:29-35), such as the graphics processing pipeline described in Williams. Thus, the combination is considered to suggest detecting a level of utilization greater than an over-utilization threshold for which a display rate of the graphics system is likely to be significantly decreased below a normal display rate); and

in response to detecting a level of utilization below an under-utilization threshold, selecting a lower performance level to reduce the clock rate in the graphics processor core clock domain to reduce power required by the graphics system (4:65-5:62, the variable speed clock applied to the floating point unit can easily be applied to one or more stages in a graphics processing pipeline (see 4:29-35), such as the graphics processing pipeline described in Williams).

See the rejection of claim 1 for rationale to combine Oliver with Giemborek and Williams.

Giemborek in view of Williams and further in view of Oliver doesn't describe a system and method wherein in response to detecting a level of utilization greater than **a non-zero, over-utilization threshold percentage level** for which a display rate of the graphics system is likely to be significantly decreased below a normal display rate, selecting a higher performance level to increase the clock rate in the graphics processor core clock domain; and

in response to detecting a level of utilization below **a non-zero, under-utilization threshold percentage level**, selecting a lower performance level to reduce the clock rate in the graphics processor core clock domain to reduce power required by the graphics system.

However, Gulick describes a system and method wherein the clock rate is adjusted according to two different non-zero threshold values, and further describes increasing the clock rate when the first threshold value is exceeded and decreasing the clock rate when the second threshold value is not exceeded (8:62-9:5 "It is noted that in an alternative embodiment a plurality of threshold flags are used. A first threshold flag may indicate an upper bound of the desired clock rate and a second threshold flag may indicate a lower bound of the desired clock rate. If the data in the buffer causes the first threshold flag to be asserted, the frame clock rate is decreased. If the data in the buffer causes the second threshold flag to be asserted, the rate of the frame clock is increased. If the data in the data buffer remains at a level between the threshold flags, the clock adjustment is successful and no adjustment is necessary.").

All the elements of claim 21 are known in Giemborek, Williams, Oliver, and Gulick, the only difference is the combination of known elements into a single system and method.

Thus, it would have been obvious to one of ordinary skill in the art at the time of the invention to include in Giemborek, Williams, and Oliver the system and method wherein the clock rate is adjusted according to two different threshold values, wherein in response to detecting a level of utilization greater than **a non-zero, over-utilization threshold percentage level** for which a display rate of the graphics system is likely to be significantly decreased below a normal display rate, selecting a higher performance level to increase the clock rate in the graphics processor core clock domain; and in response to detecting a level of utilization below **a non-zero, under-utilization threshold percentage level**, selecting a lower performance level to reduce the clock rate in the graphics processor core clock domain to reduce power required by the graphics system, as suggested by Gulick, as this doesn't change the overall operation of the system, and it could be used to achieve the predictable result of creating a small percentage range of an acceptable amount of stall cycles in the system, which has the advantage of reducing power consumption, as the clock speed will not be increased for very brief stall periods, while still not affecting perceivable system performance, as a user will not be aware of the minimal and very brief delay that occurs within the pipeline.

RE claim 31, Giemborek, Williams, and Oliver don't describe but Gulick suggests the method of claim 21, wherein the non-zero, over-utilization threshold percentage level is different from the non-zero, under-utilization threshold percentage level (see the rejection of claim 21, the two threshold levels are different).

See the rejection of claim 21 for rationale.

Claims 28 and 30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Giemborek et al. (US 6,950,105) in view of Williams et al. (US 6,397,343) in view of Oliver et al. (US 7,243,217), as applied to claims 1 and 25 above, and further in view of Culbert et al. (US 6,820,209).

As to claims 28 and 30, Giemborek describes a 2D/3D graphics engine that is capable of operating at different clock speeds (column 2 lines 42-67), where the 2D/3D graphics engine is operated at a speed that is determined based on factors that include the software running on a host CPU, as well as display mode settings of the computer, such as screen resolution, pixel or color depth, and screen refresh rate (column 5 lines 13-48).

Giemborek in view of Williams and further in view of Oliver doesn't describe the method of claims 1 or 25 wherein said at least two discrete performance levels include a low power two-dimensional graphics performance level, a standard two-dimensional graphics performance level, a low power three-dimensional graphics performance level, and a high performance three-dimensional graphics performance level.

However, Culbert describes a system and method wherein at least two discrete performance levels include a low power two-dimensional graphics performance level, a standard two-dimensional graphics performance level, a low power three-dimensional graphics performance level, and a high performance three-dimensional graphics performance level (column 5 lines 46-67 through column 6 lines 1-43 describes a system that includes a 2D graphics engine 212 and a separate 3D graphics engine 214. The 2D and 3D graphics engines are only activated when the graphics controller needs to produce 2D or 3D graphics. Further, column 6 lines 44-67 through column 7 lines 1-16 describes that the graphics controller supplies a clock signal to the 2D engine and a second clock signal to the 3D engine. To reduce power consumption, the clock signal normally sent to the 2D engine is stopped when the 2D engine is not being utilized, and likewise the clock signal normally sent to the 3D engine is stopped when its processing resources are not being utilized. Thus, Culbert is considered to describe a low power two-dimensional graphics performance level (when the 2D processor isn't being utilized), a standard two-dimensional graphics performance level (when the 2D processor is being utilized), a low-power three-dimensional graphics performance level (when the 3D processor isn't being utilized), and a high performance three-dimensional graphics performance level (when the 3D processor is being utilized)).

All the elements of claims 28 and 30 are known in Giemborek, Williams, Oliver, and Culbert, the only difference is the combination of known elements into a single system and method.

Thus, it would have been obvious to one of ordinary skill in the art at the time of the invention to include in Giemborek in view of Williams and further in view of Oliver the system and method wherein said at least two discrete performance levels include a low power two-dimensional graphics performance level, a standard two-dimensional graphics performance level, a low power three-dimensional graphics performance level, and a high performance three-dimensional graphics performance level, as taught by Culbert, as a system and method wherein the 2D engine is separate from the 3D engine, and where each engine is separately controlled, could be used to achieve the predictable result of more effectively saving power, as when only the 2D or only the 3D engine is required for a particular operation, the other engine can be powered down, which doesn't degrade the performance of the system, and reduces power consumption even more than clocking the 2D and 3D engines up and down together (as described in Giemborek).

Claim 29 is rejected under 35 U.S.C. 103(a) as being unpatentable over Giemborek et al. (US 6,950,105) in view of Williams et al. (US 6,397,343) in view of Oliver et al. (US 7,243,217), in view of Gulick (US 6,061,802), as applied to claim 21 above, and further in view of Culbert et al. (US 6,820,209).

See the rejection of claims 28 and 30 for cited portions of Culbert and rationale, as they are equally applicable to claim 29.

Conclusion

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to DANIEL WASHBURN whose telephone number is (571)272-5551. The examiner can normally be reached on 9:30 A.M. to 6 P.M..

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ulka Chauhan can be reached on 571-272-7782. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/DANIEL WASHBURN/
Examiner, Art Unit 2628
6/15/10